



THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### Plan proximity: an enhanced metric for plan stability

**Citation for published version:**

Patron, P & Birch, A 2009, Plan proximity: an enhanced metric for plan stability. in *Workshop on Verification and Validation of Planning and Scheduling Systems, 19th International Conference on Automated Planning and Scheduling (ICAPS'09)*. pp. 74-75. <<http://icaps09.uom.gr/workshops/workshops.htm>>

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Publisher's PDF, also known as Version of record

**Published In:**

Workshop on Verification and Validation of Planning and Scheduling Systems, 19th International Conference on Automated Planning and Scheduling (ICAPS'09)

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# Plan proximity: an enhanced metric for plan stability

**Pedro Patrón**

Ocean Systems Laboratory  
Heriot-Watt University, UK  
p.patron@hw.ac.uk

**Alexandra Birch**

School of Informatics  
University of Edinburgh, UK  
a.c.birch-mayne@sms.ed.ac.uk

## Abstract

Plans executed in the real world have to adapt to the differences between the expected and the observed state of the environment where is being executed. Planning strategies dealing with this adaptation process can be compared by looking at the plan proximity between the original plan and the plans that these strategies generate to replace it. Plan proximity measures the difference between these sequential plans and the expected outcome states of these plans. We present arguments to support the claim that plan proximity is a more informed metric than plan stability in order to compare planning strategies solving these adaptation to the environment of execution.

## Introduction

Plan adaptation can be performed using different strategies. Different plan repair and re-planning techniques are being used in order to deal with a changing dynamic environment of execution.

Plan stability was proposed as a metric for comparing two plans generated with different planning strategies solving this adaptation process (Fox et al. 2006). This measure considered the missing actions from the reference and the extra actions on the test plan. However, it was relaxed on the ordering of these actions in the plan and it did not considered the outcomes of these plans.

In this paper we propose an extension of the plan stability measure that takes these two factors into account providing an enhanced vision of the difference between two plans. Our metric looks at the combined plan differences and the expected outcome state differences. For both cases, plan and outcome state, it takes the missing items from the reference and the extra items appearing in the test into account.

## Plan Difference

**Definition 1** A plan  $\pi$  is an ordered list of actions of length  $n$  that is expected to transform the state of the environment of execution from an initial state  $I$  to a final state  $G$ .

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

A reference plan  $\pi_1$  and a test plan  $\pi_2$  can be compared by looking at the different number of actions that they contain. However, the point at which these actions are performed is important during plan execution as it can drastically change the outcome of the plan. Ordering is, therefore, important when comparing two plans.

The ‘diff’ algorithm (Hunt and McIlroy 1976) solves the longest common subsequence problem, which is commonly used to find the lines that do not change between two files. This can be applied to compare two plans, finding the identically ordered items that are present between two plans. In Table 1 we can see an example of a reference and test plan with their longest common subsequence shown with aligned common items. The action missing from the test sequence, and the extra actions that the test sequence contains, are also shown.

Ref	Test	Missing	Extra
	Y		Y
A	A		
B		B	
C	C		
	Z		Z
D		D	

Table 1: A sequence of actions for a reference (A B C D) and a test (Y A C Z) plan, with their longest subsequence (A C) shown aligned, the missing actions (B D) and the extra actions (Y Z).

**Definition 2** Given a reference plan  $\pi_1$  and a test plan  $\pi_2$ , the difference between  $\pi_1$  and  $\pi_2$ ,  $D_p(\pi_1, \pi_2)$  is the number of missing actions  $m_p$  from the reference plan  $\pi_1$  and the number of extra actions  $e_p$  from a test plan  $\pi_2$  that do not appear in the longest common subsequence of actions.

$$D_p(\pi_1, \pi_2) = m_p + e_p$$

The plan difference is normalized using the sum of the number of actions of the reference plan  $n_1$  and the test plan  $n_2$ .

$$\hat{D}_p(\pi_1, \pi_2) = \frac{D_p(\pi_1, \pi_2)}{n_1 + n_2} \in [0; 1]$$

The  $\hat{D}_p$  for Table 1 is  $2 + 2/4 + 4 = 0.5$ . Table 2 shows an example where plan stability and plan difference provide different interpretations. The plan stability measure would consider these two plans to be identical, as they contain the same actions. The plan difference measure, however, takes the ordering of the actions into account and returns a result of 0.5.

Ref	Test	Missing	Extra
A		A	
B	B		
	A		A

Table 2: A sequence of actions for a reference (A B) and a test (B A) plan, with their longest subsequence (B), the missing actions (A) and the extra actions (A).

### State Difference

**Definition 3** A state  $s$  is a set of proposition facts describing a configuration of the environment of execution. Proposition facts are propositions of the domain model whose variables have been all grounded to objects in the environment of execution.

The ordering of these propositions does not change the state. If  $m$  is the total number of proposition facts in the domain of execution, a state can be represented as a binary string of length  $m$  containing 1's on the position of the string that corresponds to the proposition facts that its configuration represents. From these representation, the difference between two states can be calculated using the Hamming distance (Hamming 1950). The Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different.

**Definition 4** Given a reference state  $s_1$  and a test state  $s_2$  for a common environment of execution, the state difference can be calculated as the Hamming distance between the string representation of  $s_1$  and  $s_2$ .

$$D_s(s_1, s_2) = \sum_{i=1}^m x_i \text{ where } x_i = \begin{cases} 0 & \text{if } s_1(i) = s_2(i) \\ 1 & \text{otherwise} \end{cases}$$

The state difference is normalized using the string length  $m$ .

$$\hat{D}_s(s_1, s_2) = \frac{D_s(s_1, s_2)}{m} \in [0; 1]$$

### Plan Proximity

The proximity between a reference plan and a test plan is the combination of the plan difference and the estimated final state difference.

**Definition 5** Given a reference plan  $\pi_1$  and a test plan  $\pi_2$ , their plan proximity  $PP(\pi_1, \pi_2)$  is the normalized balanced sum of the plan difference  $D_p(\pi_1, \pi_2)$  and the state difference of the estimated final states that they are expected to produce  $D_s(G_1, G_2)$ .

$$PP_\alpha(\pi_1, \pi_2) = 1 - \alpha \cdot \hat{D}_p(\pi_1, \pi_2) - (1 - \alpha) \cdot \hat{D}_s(G_1, G_2) \in [0; 1]$$

where  $\alpha \in [0; 1]$  represents a *balance factor* between plan and state difference. The logical default value of  $\alpha$  would be 0.5. However, it will depend on the particular evaluation of plans for the different environments of execution. Plan proximity can be interpreted as the percentage of similarity between two plans. It satisfies the distance function properties. In consequence, together with the set of possible plans for an environment of execution, makes up a metric space.

### Dynamic Planning Problems

The dynamic planning problem represents the problem of adapting plans in a dynamic changing environment of execution.

**Definition 6** A dynamic planning problem is a tuple  $(I, G, \pi_0, I', G')$  where  $\pi_0$  is a plan that achieves the goal  $G$  from the initial state  $I$  and  $I'$  is a new initial state from which the new goal state,  $G'$ , must be achieved.

The Plan Proximity metric space can be used to compare different planning strategies aiming to solve the dynamic planning problem. Plans coming out of these strategies can be compared and verified against an original or reference plan (Patrón, Lane, and Petillot 2009).

**Definition 7** Given a dynamic planning problem  $(I, G, \pi_0, I', G')$ , a planning strategy,  $P_1$ , achieves greater plan proximity than a planning strategy,  $P_2$ , if the plans produced by  $P_1$  and  $P_2$ ,  $\pi_1$  and  $\pi_2$  respectively, satisfy  $PP(\pi_0, \pi_2) < PP(\pi_0, \pi_1)$ .

### Conclusion

This paper presents a novel approach towards comparing different planning strategies solving the dynamic planning problem. We use the term ‘plan proximity’ to refer to a measure of the similarity between a reference plan and a test plan. This distance-based metric considers actions missing from the reference plan, extra actions added in the test plan, sequential ordering of the plans and the expected outcomes states of these plans. By taking all these factors into account, it can be claimed that plan proximity is more informed than plan stability in providing an enhanced view of comparing planning strategies in a dynamic environment.

### References

- Fox, M.; Gerevini, A.; Long, D.; and Serina, I. 2006. Plan stability: Replanning versus plan repair. In *Int. Conf. on Automated Planning and Scheduling*.
- Hamming, R. W. 1950. Error detecting and error correcting codes. *Bell System Technical Journal* 29:147–160.
- Hunt, J. W., and McIlroy, M. D. 1976. An algorithm for differential file comparison. *Computing Science Technical Report*, Bell Laboratories 41.
- Patrón, P.; Lane, D.; and Petillot, Y. 2009. Continuous mission plan adaptation for autonomous vehicles: balancing effort and reward. In *4th Workshop on Planning and Plan Execution for Real-World Systems*, 19th *Int. Conf. on Automated Planning and Scheduling*.